



2022 年（第 15 届） 中国大学生计算机设计大赛

人工智能实践赛作品报告

作品编号： 2022036960

作品名称： 智能识别预警系统

作 者： 程梁 曹云杰

填写日期： 2022 年 5 月 1 日

目 录

第 1 章 作品概述.....	1
1.1 应用方向.....	1
1.2 主要功能与特色.....	1
第 2 章 问题分析.....	1
2.1 问题来源.....	1
2.2 现有解决方案.....	2
2.3 解决问题的思路.....	2
2.3.1 数据识别与获取.....	2
2.3.2 数据管理	2
第 3 章 设计方案.....	3
3.1 总体介绍.....	3
3.2 数字识别与指针识别理论支撑	4
3.2.1 降噪-滤波算法	4
3.2.2 边缘检测算法.....	5
3.2.3 Hough 圆检测和直线检测算法	7
3.2.4 模板匹配算法.....	8
3.3 管理员可视化页面设计	9
3.4 员工可视化页面设计	9
第 4 章 系统实现.....	10
4.1 数字识别技术实现	10
4.2 指针识别技术实现	14
4.3 管理员可视化界面实现	15
4.4 员工可视化界面实现	17
第 5 章 测试分析.....	18
5.1 数字识别测试.....	18
5.2 指针表的识别.....	20
5.3 管理员后端对于表盘的实时监测	22
第 6 章 作品总结.....	23
6.1 作品特色与创新点	23
6.2 应用推广.....	23
6.3 作品展望.....	24
参考文献.....	24

第1章 作品概述

1.1 应用方向

应用于生活或者工业中对于表盘的监测管理，解决传统过程中需要专业工人在固定时间去检查表盘的难题。通过物联网及人工智能相结合，并且配备相应的管理系统，实现对表盘的实时监控，实时预警。

1.2 主要功能与特色

本项目通过树莓派的摄像头传输图像给服务端，通过相应算法识别出指针式和数字式两类表盘的读数，并按照相应的预警值发出异常警报，将表盘的实时信息发送到管理员网页和员工手机微信小程序，实现对这两类表盘的远程管理与实时监控，极大的提高了对表盘的管理与维护效率。

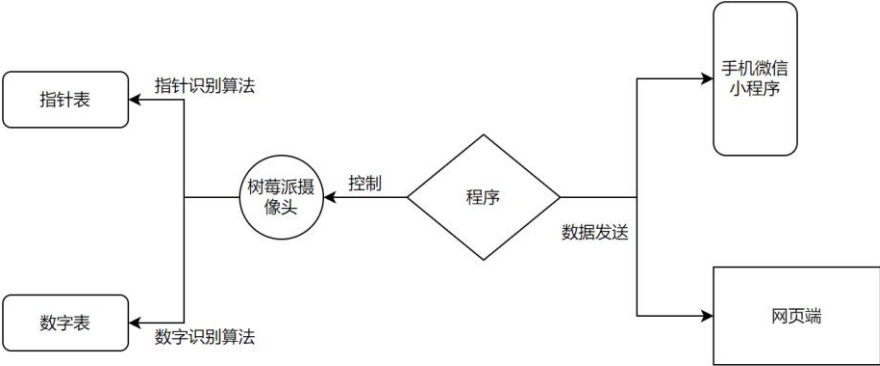


图 1 整体概述图

第2章 问题分析

2.1 问题来源

传统人工抄表的管理结构和管理方式较为落后。工人去抄表，查看表读数，

费时费力，还要花费大量时间前往目的地，还有人工抄表速度慢，误差大，特殊场景下抄表困难等情况。当表盘发生异常也不能及时知道，导致维修也有很大的延迟，造成不必要的经济损失。

2.2 现有解决方案

通过在表盘上安装摄像头进行图像数据采集，再进行网络传输到计算机上，用相应算法识别出读数，按照统一的数据结构进行保存并分析，如果达到了相应的预警值则向管理者发出异常警报，以达到精确的对表盘进行远程管理与实时监控。

2.3 解决问题的思路

2.3.1 数据识别与获取

传统表盘一般有以下分类：

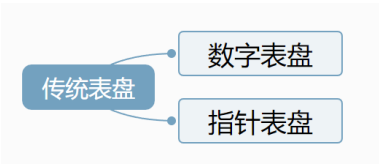


图 2 传统表盘分类图

则可以从数字识别与指针识别两个方向研究。

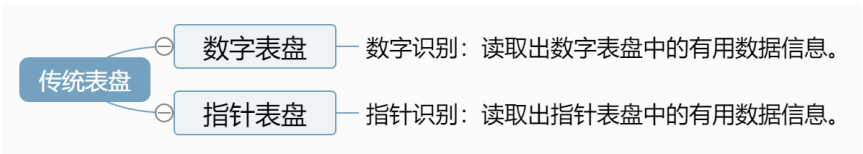


图 3 识别分类图

2.3.2 数据管理

数据的管理分为管理员层与员工层。每个管理员能够管理所有的员工信息与表盘信息，每个员工能够管理指定数量的表盘。这样的分层管理方式，能够把工作细化，提高工作效率。

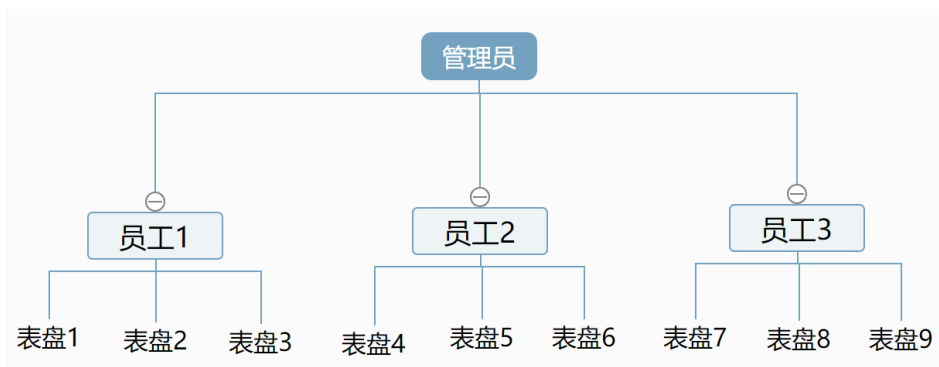


图 4 数据管理结构图

第3章 设计方案

3.1 总体介绍

通过树莓派的摄像头采集图像数据，传输给服务端，然后一系列的图像处理算法识别出指针式和数字式两类仪表的读数，将表的实时信息发送到管理员网页和员工手机微信小程序。

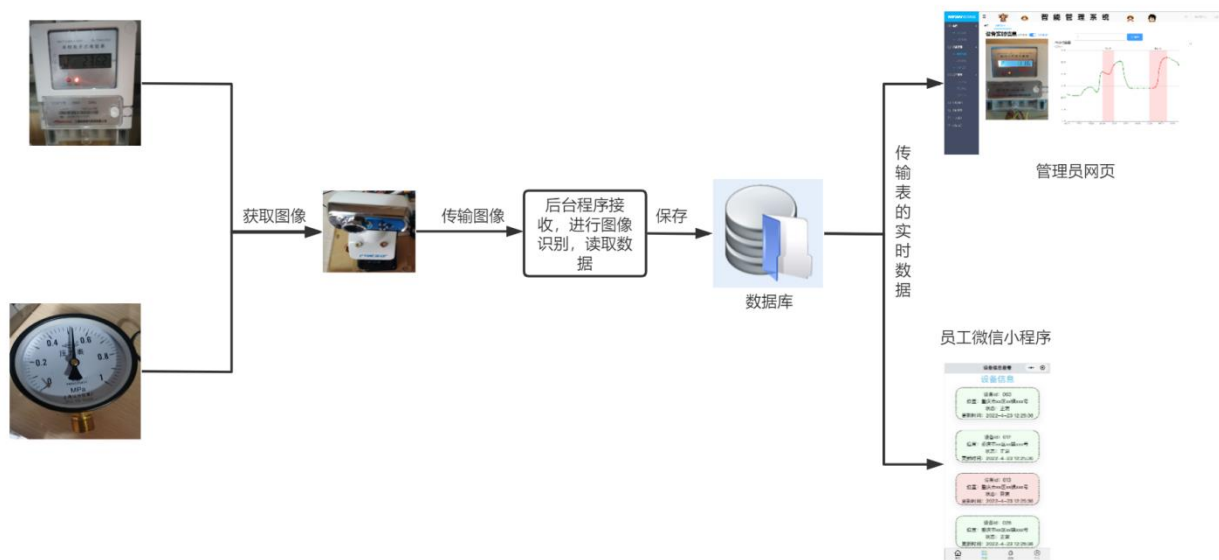


图 5 总体介绍图

3.2 数字识别与指针识别理论支撑

3.2.1 降噪-滤波算法

图像在采集、传输、接收过程，均处在复杂的外部环境中，存在各种干扰，一般都会受到噪声的影响，继而影响到输入、输出环节，从而导致图像质量下降，不仅会导致图像的分辨率降低，同时图像原本精细的结构也会遭到破坏，这样对于图像的后续处理时极为不利的，因而有效的抑制噪声，对于图像的应用有着至关重要的作用。

不同的滤波器有着截然不同的效果，经过测试，最终选择了效果较好的高斯滤波器。

高斯滤波是一种线性平滑滤波，适用于消除高斯噪声，广泛应用于图像处理的减噪过程。高斯滤波就是对整幅图像进行加权平均的过程，每一个像素点的值，都由其本身和邻域内的其他像素值经过加权平均后得到。高斯滤波的具体操作是：用一个模板（或称卷积、掩模）扫描图像中的每一个像素，用模板确定的邻域内像素的加权平均灰度值去替代模板中心像素点的值。

一维高斯分布：

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

二维高斯分布：

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

公式 1 一、二维高斯分布公式

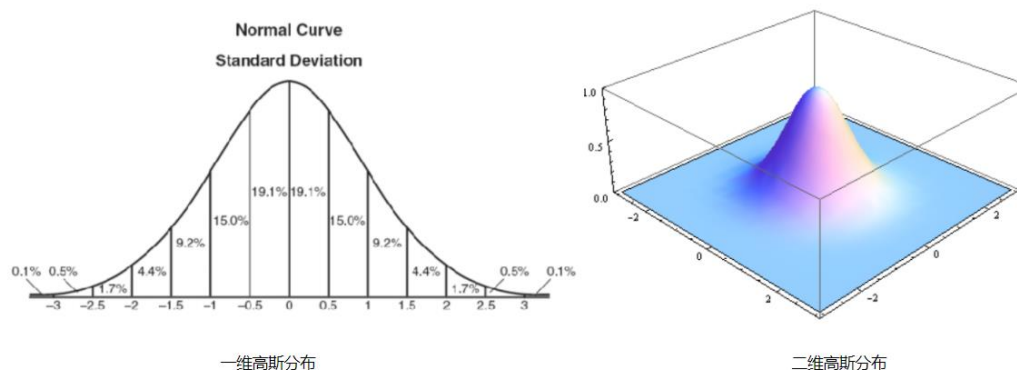


图 6 一、二维高斯公式正态分布图

通过该思想计算并且归一化得到高斯掩膜

0.075	0.124	0.075
0.124	0.204	0.124
0.075	0.124	0.075

表 1 高斯掩膜表

然后通过该掩膜对图像进行处理便可去除图像中的噪声点。

3.2.2 边缘检测算法

图像边缘是图像最基本的特征，所谓边缘(Edge) 是指图像局部特性的不连续性。灰度或结构等信息的突变处称之为边缘。例如，灰度级的突变、颜色的突变、纹理结构的突变等。边缘是一个区域的结束，也是另一个区域的开始，利用该特征可以分割图像。



图 7 理想和斜坡边缘模型图

为了达到寻找边缘的目的，检测灰度变化可用一阶导数或二阶导数来完成。下面对一阶导数进行展示：

为了在一幅图像 f 的 (x,y) 位置处寻找边缘的强度和方向，所以选择的工具就是梯度，梯度用 ∇f 来表示，并用向量来定义，定义如下所示：

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} \equiv \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

公式 2 梯度公式

梯度 ∇f 的大小用 $M(x,y)$ 表示，则：

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

公式 3 梯度大小公式

对于以为函数 $f(x)$ 在点 x 处的导数的近似：将函数 $f(x + \Delta x)$ 展开为 x 的泰勒级数，令 $\Delta x = 1$ ，且只保该级数的线性项，则函数 $f(x)$ 的梯度 ∇f 计算为：

$$\nabla f = \frac{\partial f}{\partial x} = f'(x) = f(x + 1) - f(x)$$

公式 4 函数 $f(x)$ 的梯度公式

分别计算 x 方向和 y 方向上的梯度

$$g_x = \frac{\partial f(x, y)}{\partial x} = f(x + 1, y) - f(x, y)$$

$$g_y = \frac{\partial f(x, y)}{\partial y} = f(x, y + 1) - f(x, y)$$

公式 5 函数在 x, y 方向上的梯度

常见的算子有 Roberts 算子, Prewitt 算子, Sobel 算子, Laplacian 算子，不同的算子识别效果不同。

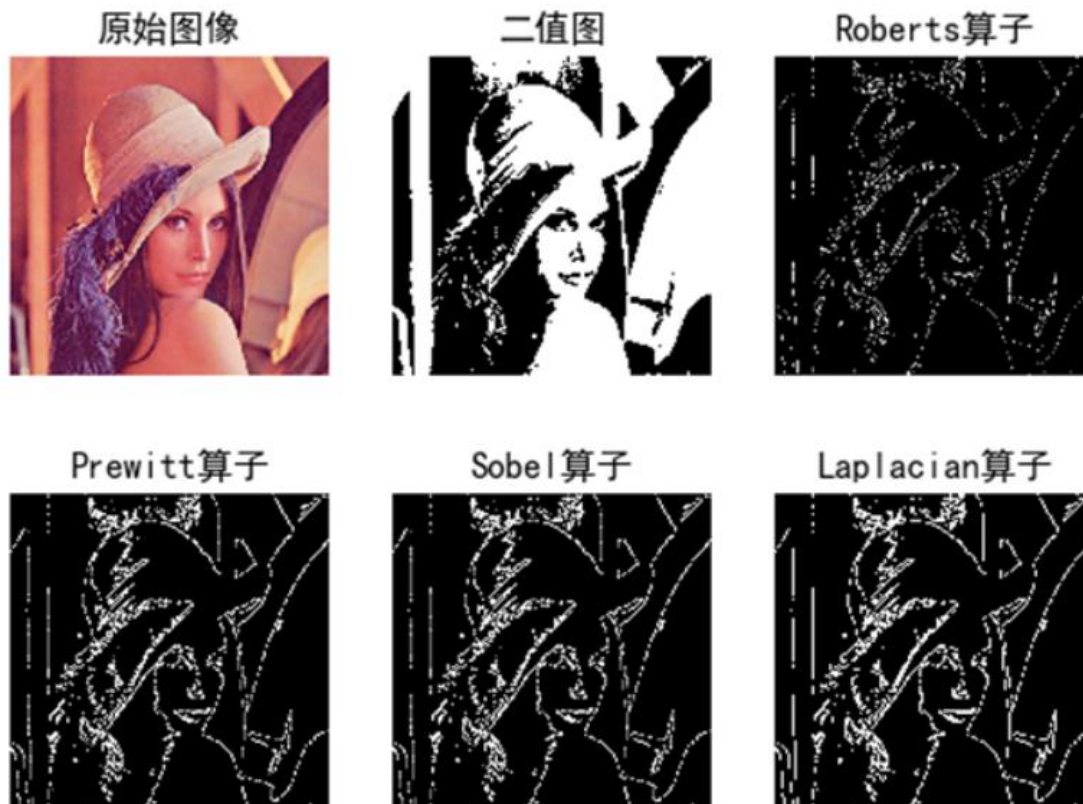


图 8 常见算子的效果图

3.2.3 Hough 圆检测和直线检测算法

Hough 将图像空间转换到参数空间，在参数空间对点进行描述，达到检测图像边缘的目的。该方法把所有可能落在边缘上的点进行统计计算，根据对数据的统计结果确定属于边缘的程度。Hough 变换的实质就是对图像进行坐标变换，把平面坐标变换为参数坐标，使变换的结果更易识别和检测。

Hough 圆检测算法的基本思路是：认为图像上每一个非零像素点都有可能是一个潜在的圆上的一点，通过投票，生成累积坐标平面，设置一个累积权重来定位圆。

标准圆公式

利用图像梯度信息的Hough变换，
对圆的标准方程对x求导得到

$$(x-a)^2 + (y-b)^2 = r^2 \longrightarrow 2(x-a) + 2(y-b)\frac{dy}{dx} = 0$$

图 9 Hough圆检测

Hough 直线检测算法就是把在图像空间中的直线检测问题转换到参数空间中对点的检测问题，通过在参数空间里寻找峰值来完成直线检测任务。

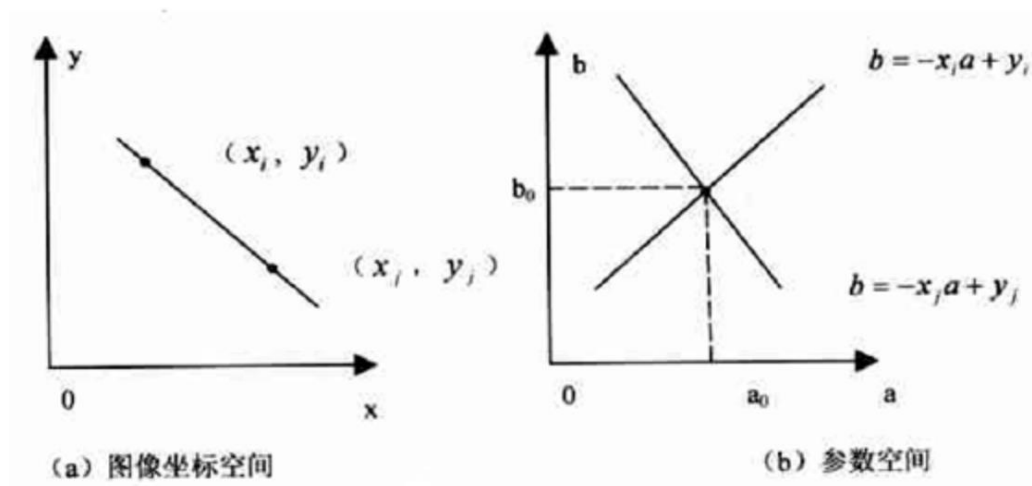


图 10 直角坐标系中的Hough变换图

图像坐标空间中过点 (x_i, y_i) 和点 (x_j, y_j) 的直线上的每一点在参数空间 a - b 上各自对应一条直线，这些直线都相交于点 (a_0, b_0) ，而 a_0 、 b_0 就是图像坐标空间 x - y 中点 (x_i, y_i) 和点 (x_j, y_j) 所确定的直线的参数。

3.2.4 模板匹配算法

模板匹配是在一幅图像中寻找一个特定目标的方法之一，使用这种方法遍历图像中的每一个可能的位置，比较各处与模板是否“相似”，当相似度足够高时，就认为找到了我们的目标。

常用的模板匹配算法公式有：

平方差匹配法 CV_TM_SQDIFF

$$R(x, y) = \sum_{x', y'} [T(x', y') - I(x + x', y + y')]^2$$

公式 6 平方差匹配公式

归一化平方差匹配法 CV_TM_SQDIFF_NORMED

$$R(x, y) = \frac{\sum_{x', y'} [T(x', y') - I(x + x', y + y')]^2}{\sqrt{\sum_{x', y'} T(x', y')^2} \cdot \sqrt{\sum_{x', y'} I(x + x', y + y')^2}}$$

公式 7 归一化平方差匹配公式

相关匹配法 CV_TM_CCORR

$$R(x, y) = \sum_{x', y'} [T(x', y') \cdot I(x + x', y + y')]$$

公式 8 相关匹配公式

归一化相关匹配法 CV_TM_CCORR_NORMED

$$R(x, y) = \frac{\sum_{x', y'} [T(x', y') \cdot I(x + x', y + y')]}{\sqrt{\sum_{x', y'} T(x', y')^2} \cdot \sqrt{\sum_{x', y'} I(x + x', y + y')^2}}$$

公式 9 归一化相关匹配公式

相关系数匹配法 CV_TM_CCOEFF

$$R(x, y) = \sum_{x', y'} [T(x', y') \cdot I(x + x', y + y')] - \frac{1}{w \cdot h} \cdot \sum_{x', y'} T(x', y') \cdot \sum_{x', y'} I(x + x', y + y')$$

公式 10 相关系数匹配公式

3.3 管理员可视化页面设计

将管理员端制作为网页，便于使用计算机对大量设备数据与员工数据的管理工作。

功能模块主要分为首页、设备管理、员工管理、发送通知、个人资料、系统设置 6 个模块：

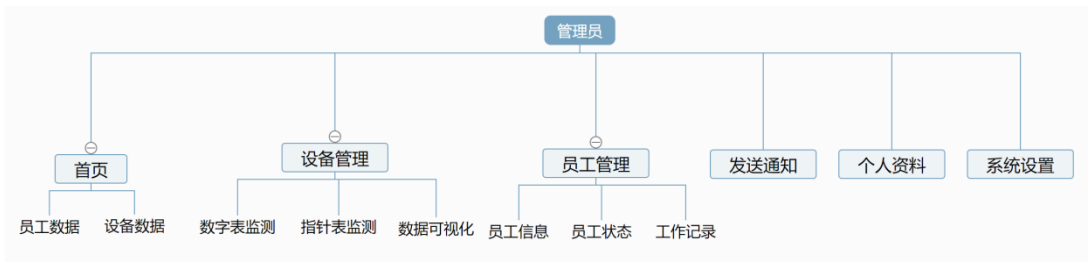


图 11 管理员端功能模块结构图

3.4 员工可视化页面设计

员工端设置为微信小程序，便于随时随地查看自己管理的表设备的实时情况，还能随时接收通知与警报。

功能模块主要分为管理、消息通知、个人 3 个模块。

具体设计如下：

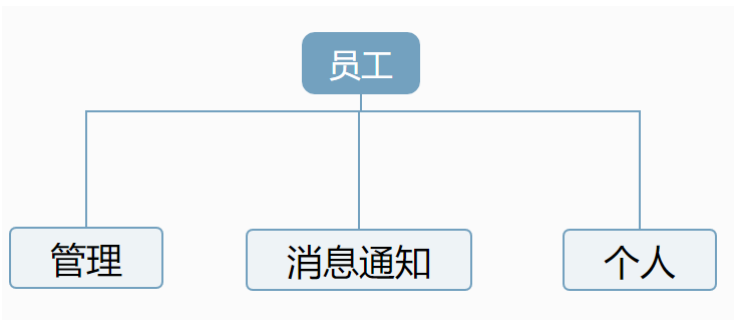


图 12 员工端功能模块结构图

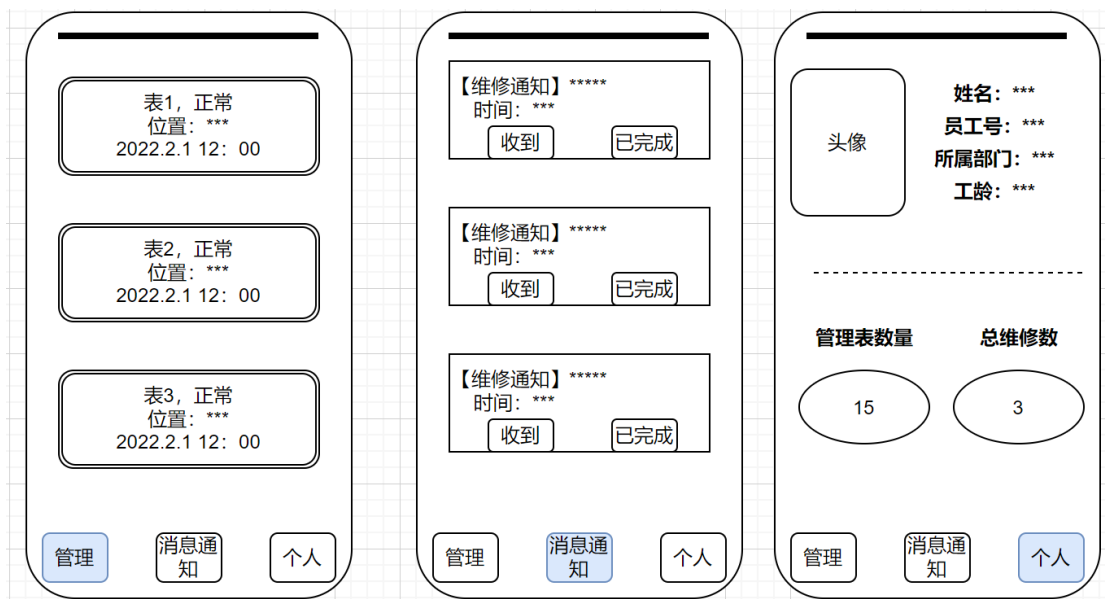


图 13 员工端页面概念设计图

第4章 系统实现

4.1 数字识别技术实现

首先将获取到的图像进行灰度处理、二值化处理，再用边缘检测、外接矩形处理识别出有效部分。



图 14 数字表原图



图 15 数字表灰度处理图

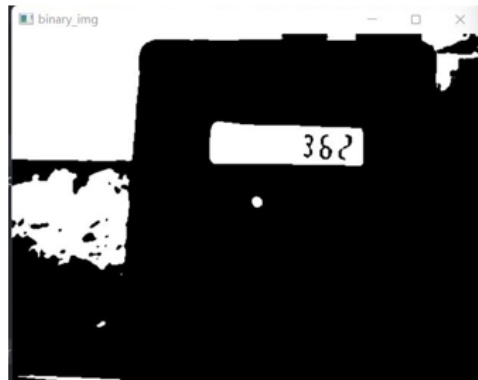


图 16 数字表二值化图



图 17 数字表有效部分识别图

接下来以小数点为分割线将有效部分中的数字分割为左右两部分，分别进行识别。

左边：将从图像中提取出来的左边部分进行二值化处理、腐蚀处理、边缘检测处理、外接矩形处理。得到如下结果：



图 18 数字表左边部分处理过程图

右边：将从图像中提取出来的右边部分同样进行二值化处理、腐蚀处理、边缘检测处理、外接矩形处理。得到如下结果：

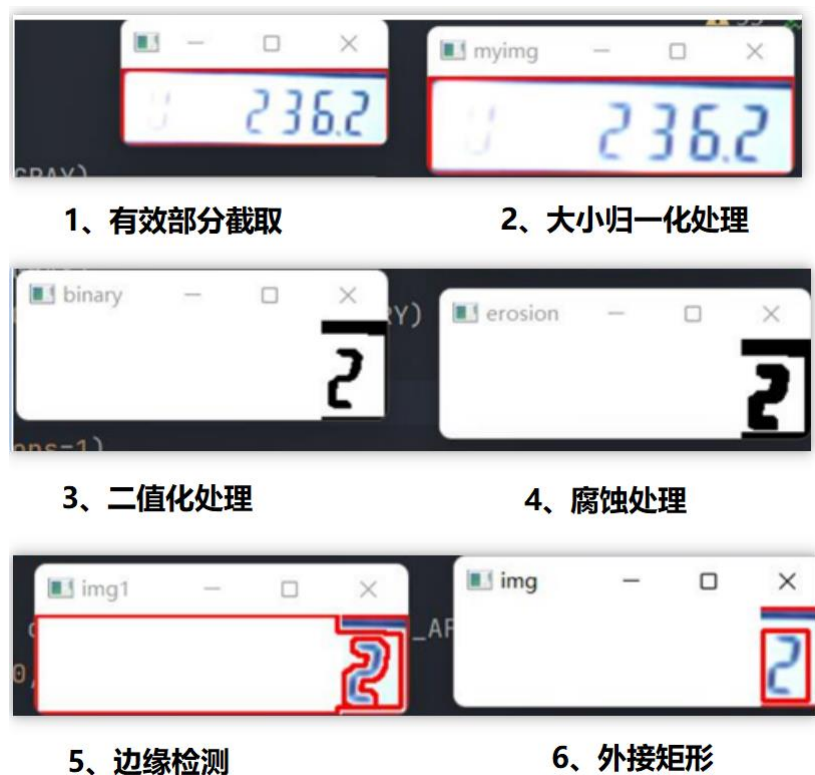


图 19 数字表右边部分处理过程图

然后将左右两边的识别结果拼接起来，进行二值化处理。这样就除去了小数点带来的干扰：



图 20 左右拼接、二值化处理图

最后将每一个数字图片进行模板匹配，得出识别结果：



图 21 匹配结果图

输出程序最终识别结果图片：



图 22 数字表识别效果图

4.2 指针识别技术实现

首先将获取到的图像进行灰度处理:

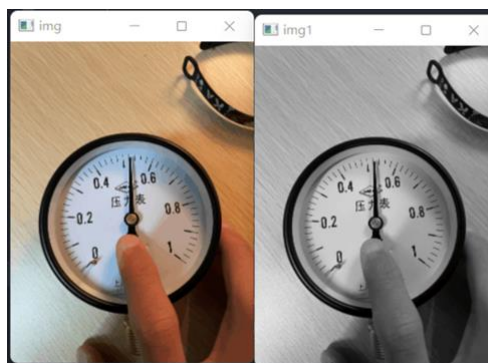


图 23 指针表原图、灰度处理图

然后进一步将图像进行二值化处理，确定圆心位置，并标记出圆心位置，这样就确定了圆形表盘的位置：

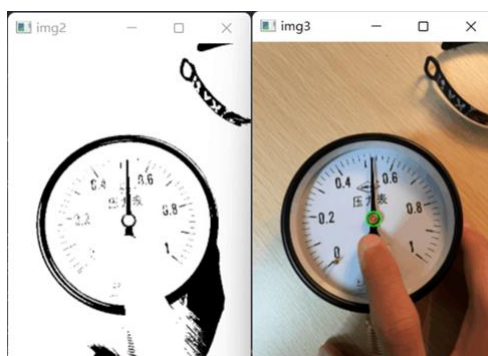


图 24 指针表二值化、圆心定位图

通过模板匹配寻找零点，然后通过零点坐标和圆心坐标就能画出起始指针



图 25 模板匹配定位原点图

接着将图像进行边缘检测处理，去噪处理，二值化处理，识别出指针直线位置并与圆心、起始直线形成一个角，最后按找角度数值占比计算出具体读数进行

输出：

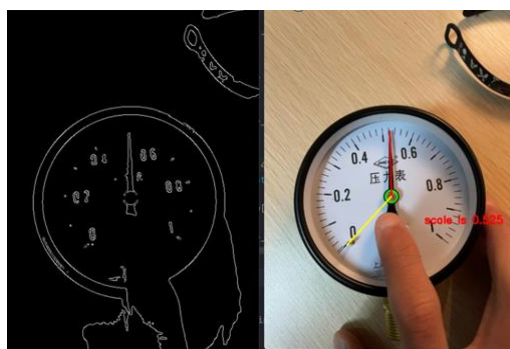


图 26 指针位置识别、最终结果图

4.3 管理员可视化界面实现

对于网页端显示，我们使用了 `html`、`css`、`javascript` 技术对管理员网页端页面进行可视化及交互设计，同时，为了扩展前端功能，增强可维护性和可扩展性，我们又使用 `layui`、`vue` 这 2 个国产框架来对网页进行布局管理，使用 `bootstrap`、`elementUI` 这些前端开发组件来对页面进行美化，简化开发过程，使代码简洁、直观、方便维护。同时也使用 `echarts`，百度地图 `api` 来对数据进行可视化。

数据存储方面，我们使用 `mysql` 数据库来进行大量数据的存储。有使用阿里巴巴的 `Druid` 数据库连接技术来对数据库进行查询访问，解决大量基于时序的数据进行聚合查询问题。

在后端开发方面，我们使用了 `spring`、`spring MVC`、`MyBatis-Plus`、`Spring Boot` 这些开发框架进行 `Java` 后端的程序开发。开发过程中始终贯彻 `ioc`，`aop` 的思想，使程序具有良好的可扩展性和可维护性。下面为项目的后端文件夹结构：

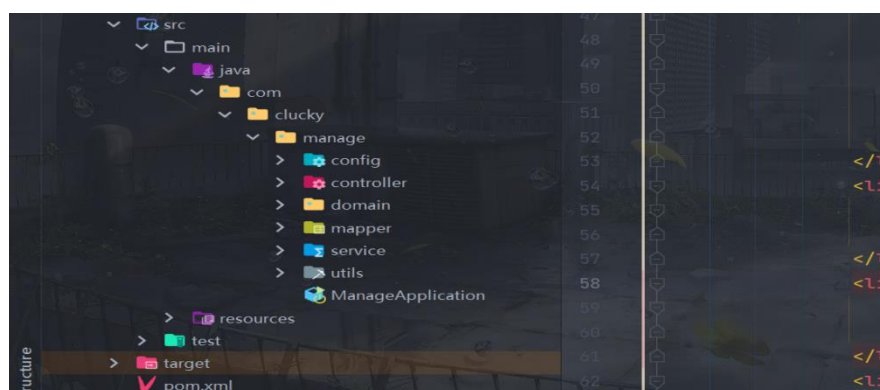


图 27 后端文件夹结构图

前后端交互使用的则是 `Axios`，`Axios` 是一个基于 `promise` 的 `HTTP` 库，异

步的进行请求，极大的增强了使用者的体验。

服务器我们则是选用了 tomcat，tomcat 不仅开源免费，而且性能强大。最终，我们做的项目部署在了 linux 系统中。

下面展示网站的最终效果图：



图 28 表盘数据统计页面图



图 29 数据可视化相关页面

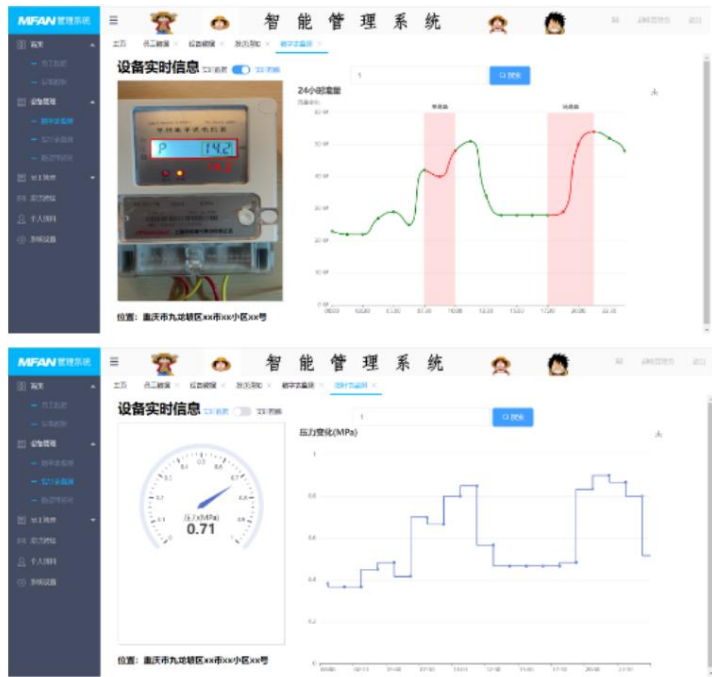


图 30 设备实时信息显示页面

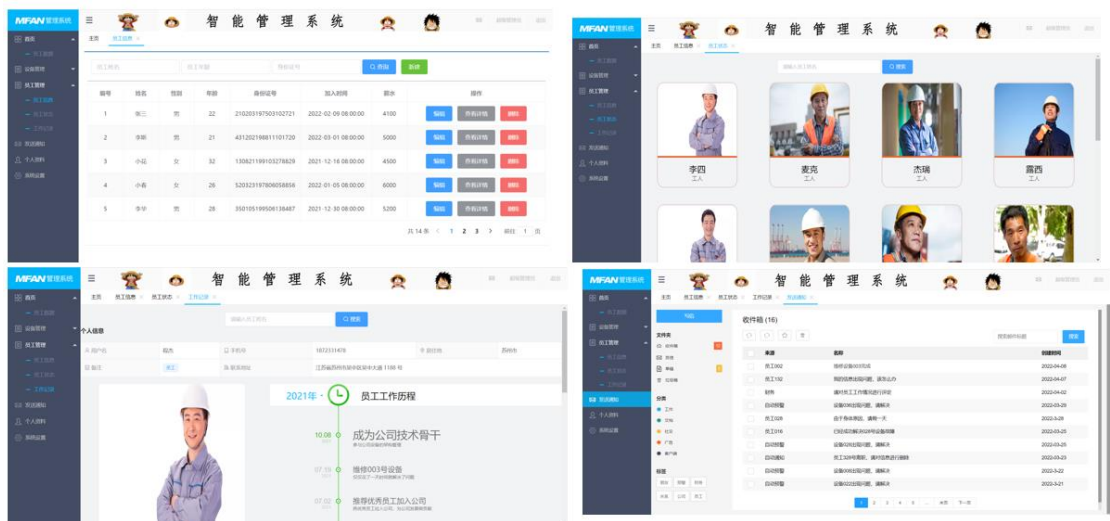


图 31 员工管理相关页面

4.4 员工可视化界面实现

使用微信开发者工具的模拟器、调试器、编辑器三个模块进行微信小程序开发页面截取如下：

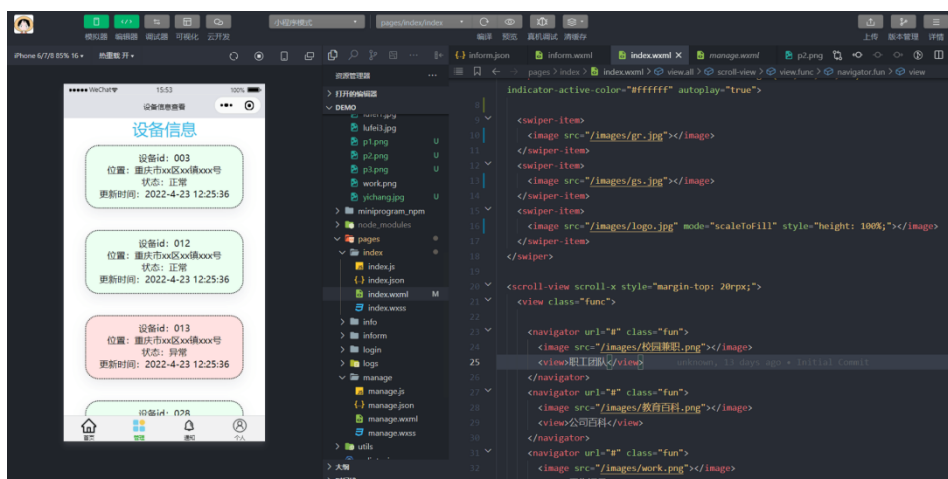


图 32 微信小程序开发页面图

最终实现，在原有设计基础上增加了登录界面和首页模块。



图 33 微信小程序效果图

第5章 测试分析

5.1 数字识别测试

上面对数字识别的过程进行了说明，现在我们对实时变化的电表进行数字的检测，下面展示部分效果图，检测的具体视频也已经给出，放在视频文件夹中，下面展示视频中的部分截图（左边的为电表，右边为识别效果）

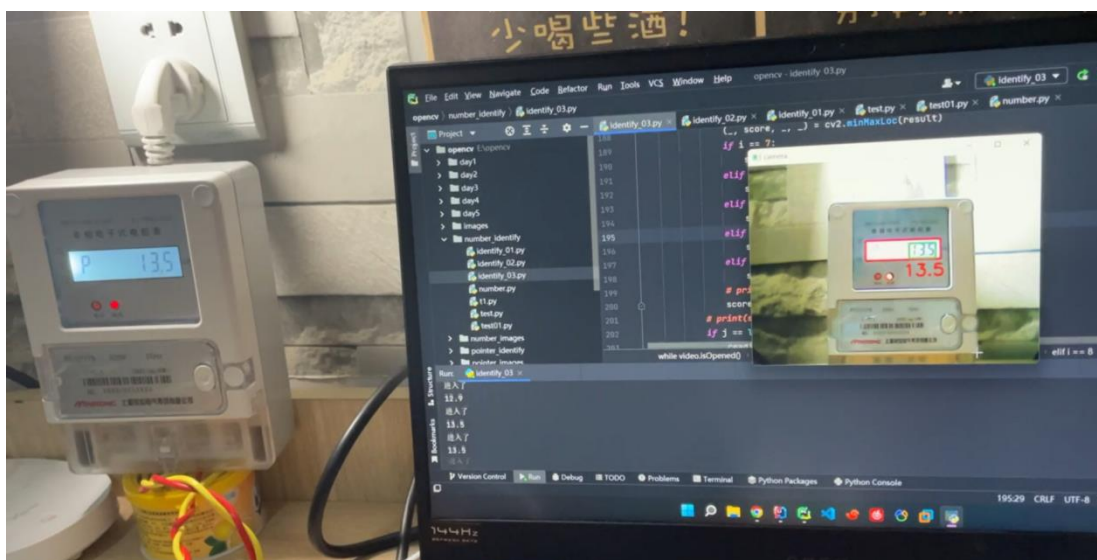


图 34 数字识别测试1

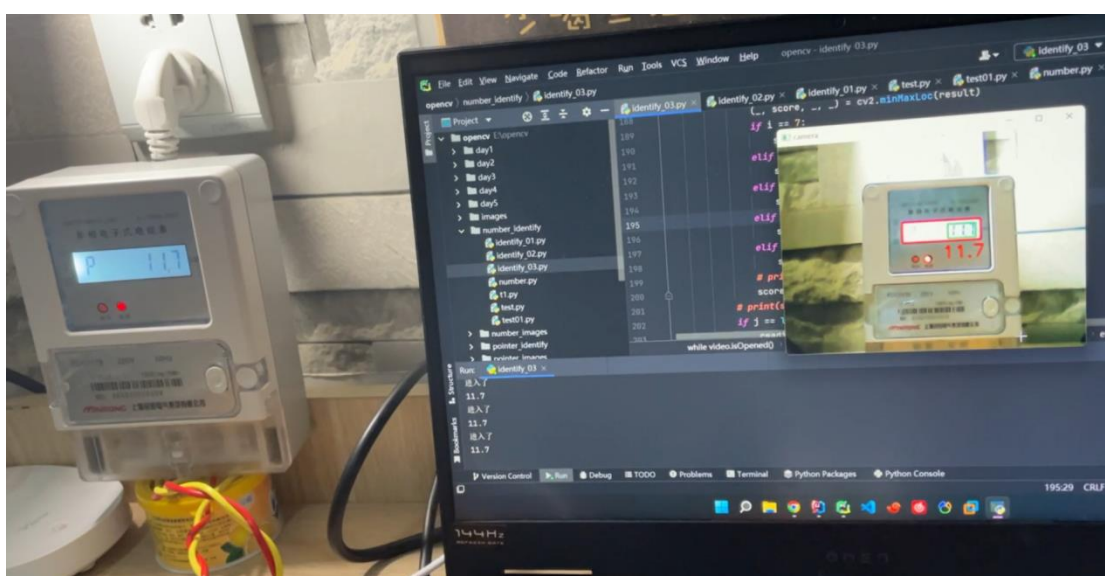


图 35 数字识别测试2

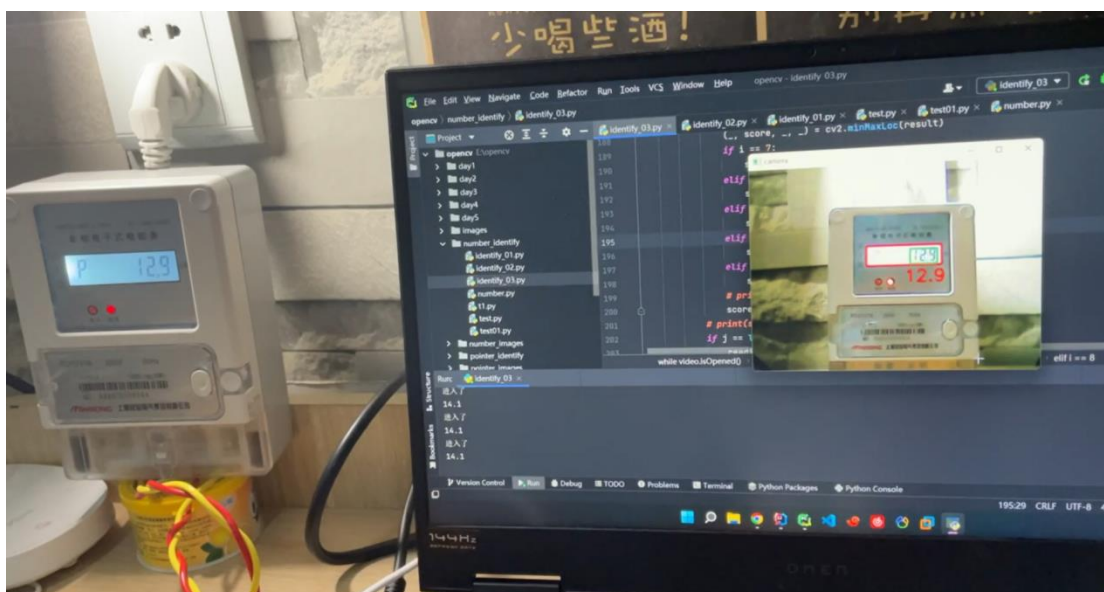


图 36 数字识别测试3

5.2 指针表的识别

指针表的识别也和数字表一样，上面已经给出了识别方法，下面来对识别的准确率进行验证，由于我们的指针表测量的是气压，我们没有气压的设备，所以我们用手来让指针进行改变，然后进行指针的识别。检测的视频也会放在视频文件夹中，下面展示部分截图：

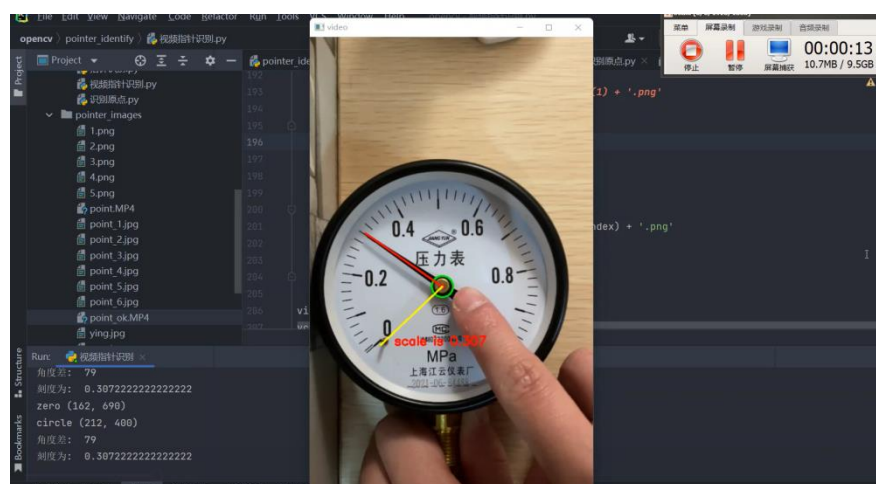


图 37 指针识别测试1

5.3 管理员后端对于表盘的实时监控

我们在通过算法将图像成功识别后，将会向服务器发出请求，将实时的图片进行发送到服务器，然后服务器会将最新的图片进行展示，每一秒更新一次，下面展示部分截图，为了方便演示，我将原本的视频放到和网页的同一个页面，左边为网页展示的图片，右边为识别出来的视频。

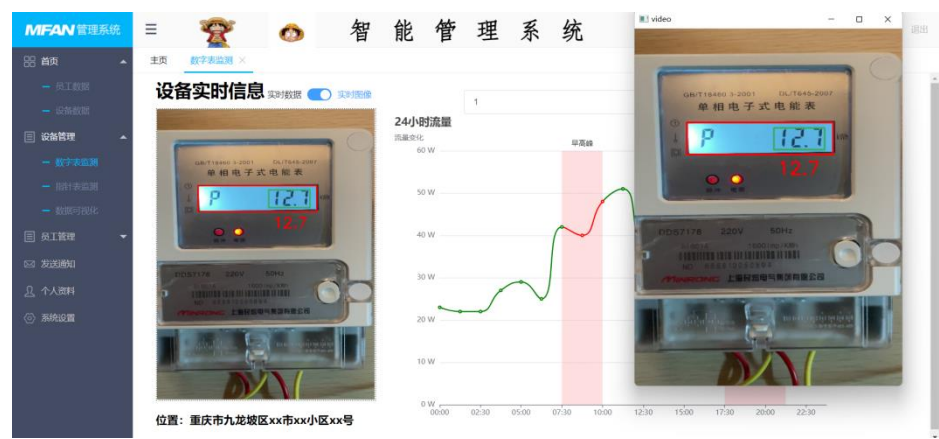


图 41 管理员端设备检测效果图1

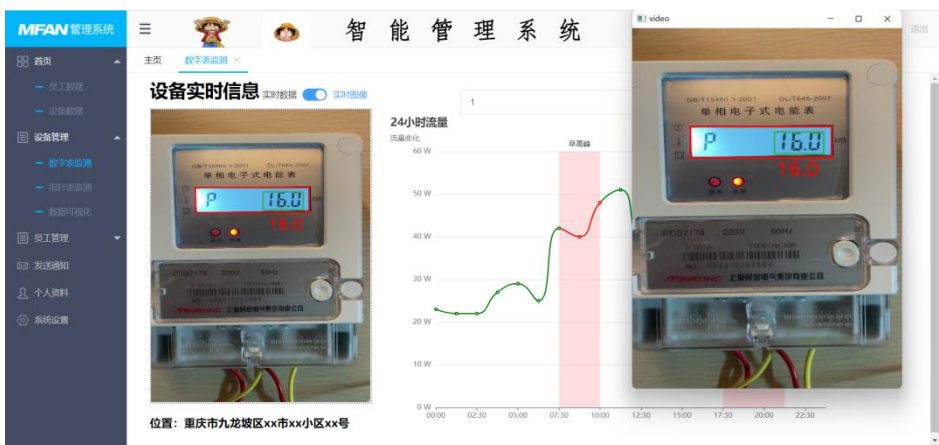


图 42 管理员端设备检测效果图2

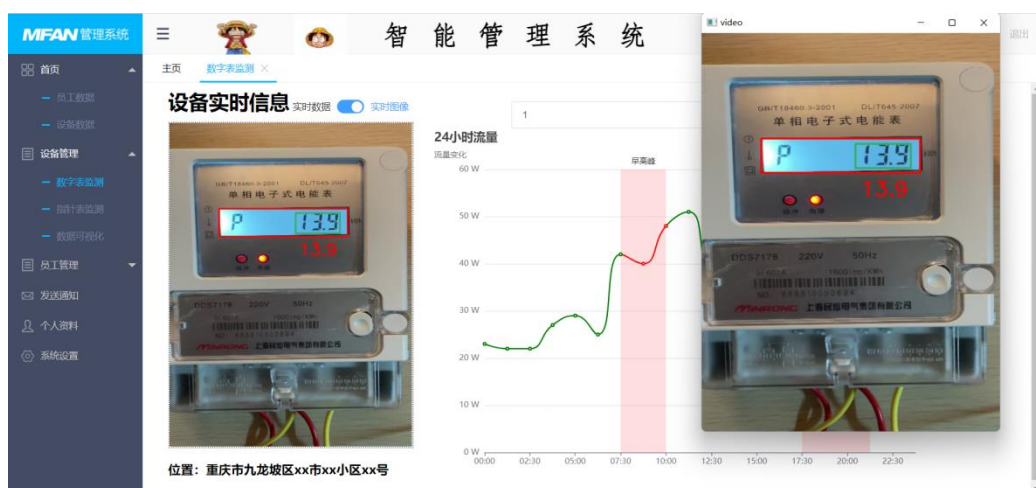


图 43 管理员端设备检测效果图3

第6章 作品总结

6.1 作品特色与创新点

我们的“智能识别预警系统”在近年来计算机技术、人工智能技术正在飞速发展，图像识别技术更是应用广泛的时代背景下诞生。

在传统的仪表盘中，如水表，需要专门的人员来对水表的读数进行读取，耗时费力。我们项目表盘进行智能化升级，实现对表盘的实时查看，当表盘读数异常也能够立马监测出来，进而对设备进行维护，极大提高了工作的效率。

它更改了传统的人工抄表的管理结构和管理方式，使用了管理员、员工的分层管理结构，利用前后端编程技术、人工智能技术、图像识别技术实现了对大量表数据的管理。能够安全，准确的监管表设备，充分的发挥图像识别技术在社会经济中的价值。

6.2 应用推广

我们的“智能识别预警系统”不仅能应用于日常生活中水表、电表等表盘仪器，还可以拓展到医学领域、工业领域、自动驾驶领域、人机交互领域等的各类表盘仪器管理。

6.3 作品展望

在传统的仪表盘中，如水表，需要专门的人员来对水表的读数进行读取，耗时费力。我们项目表盘进行智能化升级，实现对表盘的实时查看，当表盘读数异常也能够立马监测出来，进而对设备进行维护，极大提高了工作的效率。

随着物联网逐渐深入我们的生活，万物互联是一种不可阻挡的趋势，随着万物互联，智能化也是人们所追求的，相信在不久的将来，通过智能化的管理，可以取代人们辛苦的工作。

参考文献

- [1] 基于粒子滤波的 SLAM 算法并行优化与实现[J]. 朱福利,曾碧,曹军. 广东工业大学学报. 2017(02)
- [2] 一种基于模糊增强的多尺度边缘检测算法[J]. 高丽,令晓明. 兰州交通大学学报. 2008(04)
- [3] 一种基于边缘直线检测的矩形提取方法[J]. 卜飞宇. 电脑知识与技术. 2017(31)
- [4] 基于随机 Hough 变换改进的快速圆检测算法[J]. 朱正伟,宋文浩,焦竹青,郭晓. 计算机工程与设计. 2018(07)
- [5] 图像特征点匹配的流程和应用发展研究[J]. 苏子洋. 内蒙古科技与经济. 2020(07)
- [6] 基于最小二乘修正的随机 Hough 变换直线检测[J]. 乔寅骐,肖健华,黄银和,尹奎英. 计算机应用. 2015(11)
- [7] 基于减法聚类的网格霍夫变换[J]. 梁勇强,赵军,蒙峭缘. 图学学报. 2016(03)
- [8] 基于改进的概率 Hough 变换的直线检测优化算法[J]. 刁燕,吴晨柯,罗华,吴必蛟. 光学学报. 2018(08)
- [9] 快速随机 Hough 变换多圆检测算法[J]. 蒋联源,苏勤,祝英俊. 计算机工程与应用. 2009(17)
- [10] 广义 Hough 变换:多个圆的快速随机检测[J]. 黎自强,滕弘飞. 计算机辅

助设计与图形学学报. 2006(01)

[11] 基于 Canny 边缘检测和 Harris 角点检测的图像拼接方法[J]. 王玉凡. 内蒙古科技与经济. 2019(17)

[12] 高斯-拉普拉斯边缘检测算子的扩展研究[J]. 严国萍,何俊峰. 华中科技大学学报(自然科学版). 2006(10)